IN THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims

1. (Currently Amended): A method of processing inbound and outbound frames using an offload unit [[and]] having a delegated connection table, comprising:

initializing an entry in the delegated connection table with connection state data including a connection table index corresponding to a connection selected by a TCP stack executed by a host CPU for processing by the offload unit, a sequence number, and an acknowledgement number;

~~receiving the delegated connection table index at the offload engine, which is coupled to the host CPU;~~

receiving an outbound frame having a prototype header and data for transmission from the TCP stack;

~~accessing the delegated connection table entry using the delegated connection table index;~~

segmenting the data in the outbound frame into a plurality of data segments; and

for each data segment, computing a TCP checksum, determining a transmit sequence number and a transmit acknowledgement number, updating a sequence number and an acknowledgement number stored in the delegated connection table, and outputting a frame including the TCP checksum, the transmit sequence number, the transmit acknowledgement number, and the data segment for data transmission over the selected connection ~~based on a portion of the data for transmission; and~~

~~outputting a frame that includes the TCP checksum and the portion of the data for transmission~~.

2. (Cancelled)

3. (Currently Amended): The method of claim 1, wherein the ~~frame includes a received data~~ acknowledgement number stored in the delegated connection table is

updated to be a larger one of the sequence number stored in the delegated connection table and an acknowledgement number received from the TCP stack.

4. (Currently Amended): The method of claim 3, wherein the ~~received data acknowledgement number is obtained from~~ transmit sequence number is equal to a sequence number received from the TCP stack if the sequence number received from the TCP stack is greater than the sequence number stored in the delegated connection table [[entry]].

5. (Currently Amended): The method of claim 3, wherein the ~~received data acknowledgement~~ sequence number stored in the delegated connection table is updated ~~when a data is received from the destination~~ as the acknowledgement number received from the TCP stack if the acknowledgement number received from the TCP stack is greater than the sequence number stored in the delegated connection table.

6. (Currently Amended): The method of claim 1, wherein the frame includes a TCP timestamp, and the TCP timestamp is equal to a TCP timestamp received from the TCP stack if the TCP timestamp received from the TCP stack is greater than a TCP timestamp stored in the delegated connection table ~~entry and is updated when data is received from a destination connection~~.

7-8. (Cancelled)

9. (Original): The method of claim 1, further comprising computing an IPv4 header checksum when a delegated connection is an IPv4-based connection.

10-12. (Cancelled)

13. (Currently Amended): A method of processing inbound and outbound frames using an offload unit [[and]] having a delegated connection table, ~~and producing received data acknowledgements for output to a destination~~ comprising:

initializing an entry in the delegated connection table with connection [[stat]] state data including a connection table index corresponding to a connection selected by a

TCP stack executed by a host CPU for processing by the offload unit, and a count of unacknowledged frames;

receiving an inbound TCP frame from a destination connection;

determining that the destination connection is a connection delegated for processing by the offload unit;

determining whether a sequence number in the TCP frame is consecutive relative to a sequence number stored in a delegated connection table;

processing the inbound frame at the offload unit and incrementing the count of unacknowledged frames stored in the delegated connection table; ~~if the sequence number is consecutive and uploading the inbound frame to the host CPU for processing if the sequence number is not consecutive; and~~

updating the sequence number stored in the delegated connection table; and

if the count of unacknowledged frames stored in the delegated connection table is greater than a predefined limit, transmitting a receive data acknowledgement to the destination.

14. (Currently Amended): The method of claim 13, further comprising updating the count of unacknowledged frames stored in the delegated connection table to zero after transmitting the receive data acknowledgement to the destination~~:~~

~~determining that the received sequence number is greater than a threshold; and~~

~~transmitting a receive data acknowledgement to the destination~~.

15. (Previously Presented): The method of claim 13, further comprising:

determining that a timer has expired; and

transmitting a receive data acknowledgement to the destination.

16-21. (Cancelled)

22. (Currently Amended): An apparatus for processing inbound and outbound ~~frames using an offload unit and a delegated connection table and editing outbound frames~~, comprising:

a transmit engine configured to transmit outbound frames to a destination connection;

a receive engine configured to receive inbound frames from a destination connection;

a storage unit configured with a first table for storing delegated connection identification information, a second table for storing delegated connection state information, and a third table for storing memory location information, wherein each of the first, second and third tables is decoupled from one another so as to permit concurrent access of the first, second and third tables;

means for initializing an entry in the ~~delegated connection~~ first table with connection state data including a connection table index corresponding to a connection selected by a TCP stack executed by a host CPU for offload processing ~~by the offload unit~~;

~~means for determining an IPv4 checksum;~~

means for determining a TCP checksum;

means for obtaining the connection state data for the selected connection from the second table; and

means for constructing a frame for transmission at least partially responsive to the connection state data, the frame including a transmit sequence number and a transmit acknowledgement number that have been determined based on a sequence number and an acknowledgement number stored in the second table.

23. (Currently Amended): The apparatus of claim 22, wherein the [[state]] connection state data includes a [[received]] sequence number, a TCP timestamp and ~~a received data~~ an acknowledgement number.

24-25. (Cancelled)

26. (Previously presented): The method of claim 1 wherein connections that are not delegated to the offload unit or that require special processing are processed by the TCP stack; and

the offload unit may request legacy processing of one of the delegated connections by the TCP stack so that outgoing frames are transmitted by either the TCP stack or the offload connection.

27. (Previously presented): The method of claim 26 wherein a system memory is coupled to both the host CPU and the offload unit, and including the step of storing connection data for all active connections including those processed at the offload unit and the TCP stack at the host CPU at a connection table in the system memory.

28. (Previously presented): The method of claim 26 including the step of determining that an incoming frame is one of the delegated frames; and

determining the frame type of an incoming delegated frame, an incoming frame including an IP packet with a TCP segment being transferred to the TCP stack for processing.

29. (Previously presented): The method of claim 26 including the step of determining the validity of an incoming frame and uploading an invalid frame to the TCP stack for legacy processing.

30. (Previously presented): The method of claim 27 including the step of processing a valid frame at the offload unit and loading the processed frame directly to the system memory associated with the connection table.

31. (Previously presented): The method of claim 30 wherein a valid frame is partially processed at the offload unit when user buffer space associated with the offload unit is available, the valid frame being transferred to the TCP stack for legacy processing when user buffer space is not available.

32. (Previously presented): The method of claim 31, wherein a portion of the valid frame processed at the offload unit is limited by a startup limit stored at the delegated connection table.

33. (Previously presented): The method of claim 30, wherein each incoming frame includes an acknowledgement (ACK), the offload unit coalescing a plurality of the ACK's before notifying the TCP stack of the status of the delegated connection associated with each received ACK.

34. (Previously presented): The method of claim 1, wherein the offload unit compares a sequence number (SN) in each of the incoming frames with a SN stored in the delegated connection table; and

if not equal then the entire frame is uploaded for legacy processing by the TCP stack.

35. (New): A method of processing inbound and outbound frames using an offload unit having a delegated connection table, comprising:

initializing an entry in the delegated connection table with connection state data including a connection table index corresponding to a connection selected by a TCP stack executed by a host CPU for processing by the offload unit, a sequence number, an acknowledgement number, a count of recovered frames, and a count of unacknowledged frames;

receiving an inbound TCP frame from a destination connection;

determining that the destination connection is a connection delegated for processing by the offload unit;

comparing a sequence number in the TCP frame with a sequence number stored in the delegated connection table;

if the sequence number in the TCP frame is less than the sequence number stored in the delegated connection table, transmitting a receive data acknowledgement to the destination and requesting legacy processing;

if the sequence number in the TCP frame is greater than the sequence number stored in the delegated connection table and the count of recovered frames stored in the delegation connection table is less than 3, incrementing the count of recovered frames stored in the delegation connection table, transmitting a receive data acknowledgement to the destination, and requesting legacy processing; and

if the sequence number in the TCP frame is equal to the sequence number stored in the delegated connection table and the count of unacknowledged frames stored in the delegated connection table is greater than a predefined limit, transmitting a receive data acknowledgement to the destination and updating the sequence number stored in the delegated connection table.